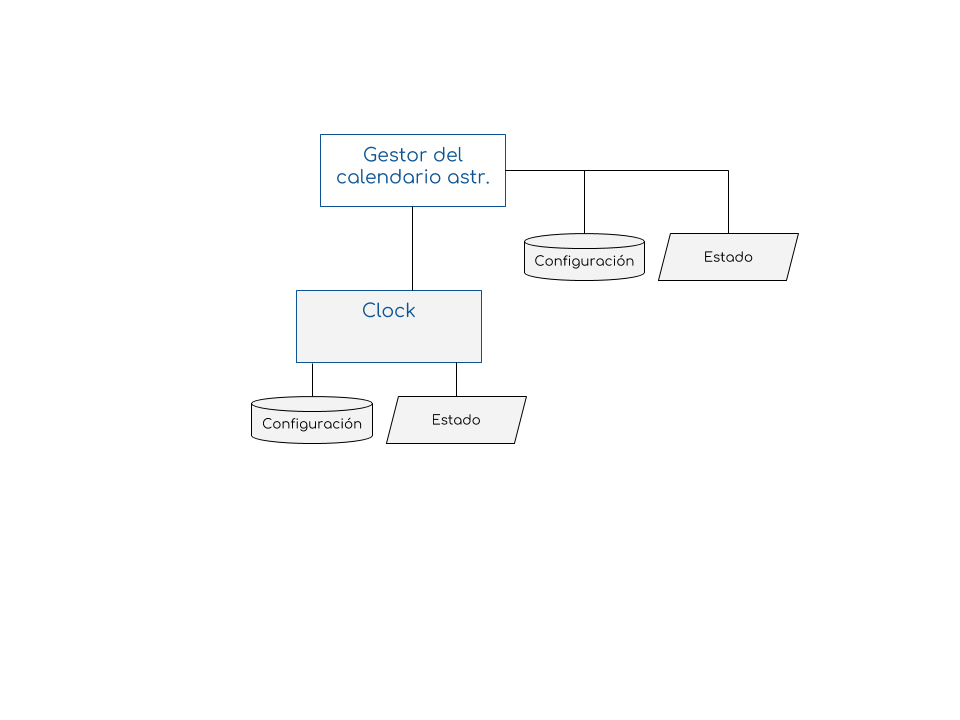
**Modelos de datos relativos al**

**Componente de Gestión del Calendario Astronómico**

El diagrama general que representa al Gestor del calendario astronómico es el siguiente:

****

**Objetos, esquemas e identificación de versiones**

Los modelos de datos se pueden organizar en base a un esquema común formado por varios nombres separados por ‘**:**’, del tipo:

**name0**:**name1**:**name2**:**…:nameN**

**-------------------**

Los objetos se representarán en formato JSON, por lo que para su identificación se utilizará un **UID** (uint32) que incluya información sobre el tipo de objeto y la versión del mismo. Este UID será utilizado en el lado del Servidor para poder catalogar el tipo de datos recibidos antes de su registro en la base de datos.

Una posible nomenclatura del UID al ser un uint32 (0xaabbccdd) podría ser ésta (que permite identificar más de 1 millón de objetos distintos con 4096 versiones diferentes para cada uno de ellos):

|  |  |
| --- | --- |
| Bits 31- 20 | Bits 19 – 0 |
| Identificador de la versión | Identificador único del tipo de objeto |

Así por ejemplo si un objeto cuyo UID = 1 (0x00000001), indicará que su versión es 0 (ya que versión 0 🡪 0x**000**00001). Mientras que ese mismo objeto, en su versión 1, tendría un UID = 1048577 (ya que versión 1 🡪 0x**001**00001).

**Gestor del calendario astronómico**

En la siguiente tabla se definen los diferentes modelos de datos relativos al módulo que gestiona la medida y que puede integrar uno o varios analizadores de diferente tipo.

|  |  |
| --- | --- |
| **Objetos JSON** | **Descripción** |
| ***calendar:manager*** | *Objeto que incluye toda la información asociada a un gestor del calendario con hitos astronómicos. Incluye información como:*   * *Identificador único que incluye información sobre el tipo de objeto y la versión del mismo* * *Objeto de gestión del tiempo real (RealTimeClock)* * *Lista de periodos temporales (festividades, periodos vacacionales, etc.)* * *Parámetros de configuración del gestor* * *Variables de estado del gestor* |

**Reloj de tiempo real**

En la siguiente tabla se definen los diferentes modelos de datos relativos únicamente al objeto que representa a un reloj de tiempo real.

|  |  |
| --- | --- |
| **Objetos JSON** | **Descripción** |
| ***calendar:clock*** | *Objeto que incluye toda la información asociada a un reloj, como:*   * *Zona horaria* * *Hora y fecha* |

**Periodo temporal**

En la siguiente tabla se definen los diferentes modelos de datos relativos únicamente al objeto que representa a un periodo de tiempo dado.

|  |  |
| --- | --- |
| **Objetos JSON** | **Descripción** |
| ***calendar:period*** | *Objeto que incluye toda la información asociada a un periodo de tiempo, como:*   * *Hora y fecha de inicio* * *Hora y fecha de finalización* * *Flag de habilitación* |

**Zona horaria/geográfica**

En la siguiente tabla se definen los diferentes modelos de datos relativos únicamente al objeto que representa a un objeto con parámetros de geolocalización

|  |  |
| --- | --- |
| **Objetos JSON** | **Descripción** |
| ***calendar:geoloc*** | *Objeto que incluye toda la información asociada a un objeto astronómico, como:*   * *Coordenadas latitud-longitud* * *Zona horaria* |

**Resumen de objetos. Asignación de UIDs**

En la siguiente tabla se enumeran todos los objetos relativos al Gestor del calendario (***time***), asignando sus rangos UID, para su identificación en la parte del Servidor.

|  |  |
| --- | --- |
| **Objetos JSON** | **Rangos UIDs (hasta 4096 versiones por UID)** |
| ***calendar:manager*** | *0x0000000B – 0xfff0000B* |
| ***calendar:clock*** | *0x0000000C – 0xfff0000C* |
| ***calendar:period*** | *0x0000000D – 0xfff0000D* |
| ***calendar:geoloc*** | *0x0000000E – 0xfff0000E* |
| *Otros Objetos externos requeridos:* | |
|  | *0x000E0000 – 0xfffE0000* |

**Descripción de los diferentes objetos**

|  |  |
| --- | --- |
| ***calendar:manager*** | |
| ***Objeto JSON:***  *{*  *"uid": u32,*  *"cfg": {*  *"updFlags": u32,*  *"evtFlags": u32,*  *"verbosity": u8,*  *},*  *"clock": calendar:clock,*  *}* | |
|  | |
| *uid : u32*  *Read-only* | *Descripción*  *Identificador del tipo de objeto (0000B) y versión*  *Valores*  *0x0000000B – 0xFFF0000B siendo XYZ el identificador de la versión* |
| *cfg.updFlags : u32*  *Read-Write* | *Descripción*  *Conjunto de flags, que habilitan o no, la notificación de los cambios hechos en algún parámetro de configuración.*  *Valores*  *CalendarManagerCfgUpdNotif* = (1 << 0), /// Habilita notificación de cambios en cualquier parámetro de la configuración  *[default = CalendarManagerCfgUpdNotif* *]* |
| *cfg.evtFlags : u32*  *Read-Write* | *Descripción*  *Conjunto de flags, que habilitan las notificaciones de los diferentes eventos. Para que un evento sea notificado, su flag debe estar activado, en caso contrario, aunque la condición de evento se produzca, no se generará ninguna notificación.*  *Valores*  *CalendarClockNoEvents* = 0, //!< No hay eventos  *CalendarClockYearEvt* = (1 << 0), //!< Evento al cambiar de año  *CalendarClockIVEvt* = (1 << 1), //!< Evento al cambiar de invierno a verano  *CalendarClockVIEvt* = (1 << 2), //!< Evento al cambiar de verano a invierno  *CalendarClockMonthEvt* = (1 << 3), //!< Evento al cambiar de mes  *CalendarClockWeekEvt* = (1 << 4), //!< Evento al cambiar de semana  *CalendarClockDayEvt* = (1 << 5), //!< Evento al cambiar de día  *CalendarClockMiddayEvt* = (1 << 6), //!< Evento al pasar por el medio día  *CalendarClockPreDuskEvt* = (1 << 7), //!< Evento al pasar por el inicio de la ventana temporal de ocaso  *CalendarClockDuskEvt* = (1 << 8), //!< Evento al ocaso  *CalendarClockPostDuskEvt* = (1 << 9), //!< Evento al pasar por el final de la ventana temporal de ocaso  *CalendarClockReducStartEvt*= (1 << 10),//!< Evento al pasar por la hora de inicio de reducción de flujo luminoso  *CalendarClockReducStopEvt*= (1 << 11),//!< Evento al pasar por la hora de finalización de reducción de flujo luminoso  *CalendarClockPreDawnEvt* = (1 << 12),//!< Evento al pasar por el inicio de la ventana temporal de orto  *CalendarClockDawnEvt* = (1 << 13),//!< Evento al pasar por el orto  *CalendarClockPostDawnEvt* = (1 << 14),//!< Evento al pasar por el final de la ventana temporal de orto  *CalendarClockHourEvt* = (1 << 15),//!< Evento al cambiar de hora  *CalendarClockMinEvt* = (1 << 16),//!< Evento al cambiar de minuto  *CalendarClockSecEvt* = (1 << 17),//!< Evento al cambiar de segundo  *CalendarClockDawnDuskUpdEvt*= (1 << 18),//!< Evento al actualizar las horas de orto y ocaso  *CalendarClockPeriodEvt* = (1 << 19),//!< Evento al cambiar de periodo  *CalendarClockEvtINVALID* = (1 << 31),//!< Indica un evento inválido    *[default = CalendarClockMinEvt* *]* |
| *cfg.verbosity : u8*  *Read-Write* | *Descripción*  *Nivel de visualización de las trazas de depuración.*  *Valores*  *ESP\_LOG\_NONE* /\*!< No log output \*/  *ESP\_LOG\_ERROR* /\*!< Critical errors, software module can not recover on its own \*/  *ESP\_LOG\_WARN* /\*!< Error conditions from which recovery measures have been taken \*/  *ESP\_LOG\_INFO* /\*!< Information messages which describe normal flow of events \*/  *ESP\_LOG\_DEBUG* /\*!< Extra information which is not necessary for normal use (values, pointers, …). \*/  *ESP\_LOG\_VERBOSE* /\*!< Bigger chunks of information, or messages which can flood the terminal \*/    *[default = ESP\_LOG\_WARN]* |
| *clock : calendar:clock*  *Read-write* | *Descripción*  *Objeto que contiene los datos asociados al reloj de tiempo real integrado* |

|  |  |
| --- | --- |
| ***calendar:clock*** | |
| ***Objeto JSON:***  *{*  *"uid": u32,*  *"cfg": {*  *"periods": [calendar:period],*  *"geoloc": calendar:geoloc*  *}*  *"stat": {*  *"flags": u32,*  *"period": u8,*  *"localtime": time\_t (long),*  *"dawn": time\_t (long),*  *"dusk": time\_t (long),*  *},*  *}* | |
|  | |
| *uid : u32*  *Read-only* | *Descripción*  *Identificador del tipo de objeto (0000C) y versión*  *Valores*  *0x0000000C – 0xFFF0000C siendo XYZ el identificador de la versión* |
| *cfg.periods : [calendar:period]*  *Read-write* | *Descripción*  *Lista (array) de periodos especiales del calendario: festivos, vacaciones, periodo estival, invernal, etc.* |
| *cfg.geoloc : calendar:geoloc*  *Read-write* | *Descripción*  *Objeto que contiene los datos de localización geográfica para la gestión del calendario astronómico, como: latitud y longitud, zona gmt, cambios de estación, correcciones astronómicas por periodos, etc.* |
| *stat.flags : u32*  *Read-only* | *Descripción*  *Conjunto de flags, que muestran los eventos generados actualmente.*  *Valores*  *ClockNoEvents* = 0, //!< No hay eventos  *ClockYearEvt* = (1 << 0), //!< Evento al cambiar de año  *ClockIVEvt* = (1 << 1), //!< Evento al cambiar de invierno a verano  *ClockVIEvt* = (1 << 2), //!< Evento al cambiar de verano a invierno  *ClockMonthEvt* = (1 << 3), //!< Evento al cambiar de mes  *ClockWeekEvt* = (1 << 4), //!< Evento al cambiar de semana  *ClockDayEvt* = (1 << 5), //!< Evento al cambiar de día  *ClockMiddayEvt* = (1 << 6), //!< Evento al pasar por el medio día  *ClockPreDuskEvt* = (1 << 7), //!< Evento al pasar por el inicio de la ventana temporal de ocaso  *ClockDuskEvt* = (1 << 8), //!< Evento al ocaso  *ClockPostDuskEvt* = (1 << 9), //!< Evento al pasar por el final de la ventana temporal de ocaso  *ClockReducStartEvt*= (1 << 10), //!< Evento al pasar por la hora de inicio de reducción de flujo luminoso  *ClockReducStopEvt* = (1 << 11), //!< Evento al pasar por la hora de finalización de reducción de flujo luminoso  *ClockPreDawnEvt* = (1 << 12), //!< Evento al pasar por el inicio de la ventana temporal de orto  *ClockDawnEvt* = (1 << 13), //!< Evento al pasar por el orto  *ClockPostDawnEvt* = (1 << 14), //!< Evento al pasar por el final de la ventana temporal de orto  *ClockHourEvt* = (1 << 15), //!< Evento al cambiar de hora  *ClockMinEvt* = (1 << 16), //!< Evento al cambiar de minuto  *ClockSecEvt* = (1 << 17), //!< Evento al cambiar de segundo  *ClockDawnDuskUpdEvt*= (1 << 18),//!< Evento al actualizar las horas de orto y ocaso  *ClockPeriodEvt* = (1 << 19), //!< Evento al cambiar de periodo  *ClockEvtINVALID* = (1 << 31), //!< Indica un evento inválido  *[default = ClockMinEvt]* |
| *stat.period : u8 [OPCIONAL]*  *Read-only* | *Descripción*  *Identificador del periodo activo (de los configurados en la lista de periodos <cfg.periods>. En caso de que no haya ningún periodo activo, este campo puede ser omitido.*  *Valores*  *0 a NumPeriodos-1* |
| *stat.localtime : time\_t (long)*  *Read-only* | *Descripción*  *Hora local en formato time\_t.* |
| *stat.dawn : time\_t (long)*  *Read-only* | *Descripción*  *Hora de inicio del próximo orto en formato time\_t (hora local). Se actualiza cuando se produce el evento de ocaso o cuando se actualiza la configuración del objeto.* |
| *stat.dusk : time\_t (long)*  *Read-only* | *Descripción*  *Hora de inicio del próximo ocaso en formato time\_t (hora local). Se actualiza cuando se produce el evento de orto o cuando se actualiza la configuración del objeto.* |

|  |  |
| --- | --- |
| ***calendar:period*** | |
| ***Objeto JSON:***  *{*  *"uid": u32,*  *"since": time\_t (long),*  *"until": time\_t (long)*  *"enabled": boolean*  *}* | |
|  | |
| *uid : u32*  *Read-only* | *Descripción*  *Identificador del tipo de objeto (0000D) y versión*  *Valores*  *0x0000000D – 0xFFF0000D siendo XYZ el identificador de la versión* |
| *since : time\_t (long)*  *Read-write* | *Descripción*  *Hora local en formato time\_t a la que comienza el periodo.* |
| *until : time\_t (long)*  *Read-write* | *Descripción*  *Hora local en formato time\_t a la que finaliza el periodo.* |
| *enabled : boolean*  *Read-write* | *Descripción*  *Flag que activa o desactiva la ejecución del periodo.* |

|  |  |
| --- | --- |
| ***calendar:geoloc*** | |
| ***Objeto JSON:***  *{*  *"uid": u32,*  *"timezone": char[]*  *"coords": [latitude (double), longitude (double)],*  *"astCorr": [[dawnCorr (time\_t), duskCorr (time\_t)]]*  *}* | |
|  | |
| *uid : u32*  *Read-only* | *Descripción*  *Identificador del tipo de objeto (0000E) y versión*  *Valores*  *0x0000000E – 0xFFF0000E siendo XYZ el identificador de la versión* |
| *timezone : char[]*  *Read-write* | *Descripción*  *Cadena de texto que representa la zona de hora local con cambios de estación DST en formato <tzset>: "GMT±KGMT±L,Mm.s.w/h, Mm.s.w/h"*  *Valores*  *Para Madrid (ejemplo) es zona GMT+1 en invierno y GMT+2 en verano con cambios de estación el último domingo de marzo y octubre a las 02:00.*  *"GMT-1GMT-2,M3.5.0/2,M10.5.0"* |
| *coords : [double, double]*  *Read-write* | *Descripción*  *Array formado por los datos de latitud y longitud en formato decimal. Tipo de dato <double>* |
| *astcorr : [[time\_t, time\_t]]*  *Read-write* | *Descripción*  *Array con diferentes correcciones aplicables al orto y al ocaso para un ajuste fino de los mismos. Cada elemento del array es a su vez un array de 2 elementos (corrección aplicada al orto y corrección aplicada al ocaso).*  *Valores*  *Al ser utilizado en un objeto <calendar:clock>, el array suele contener datos de corrección para diferentes periodos del año y por lo tanto la longitud del array suele coincidir con el número de periodos configurados en dicho objeto <calendar:clock/cfg.periods>* |

**Otros objetos externos requeridos**